LA-UR -80-278

TITLE: GROUND BASED AUTOMATED TELESCOPE

MASTER

AUTHOR(S): Stirling A. Colgate, T-DOT
William Thompson, New Mexico Tech

SUBMITTED TO: Presentation at Optical and Infrared Program,
Tucson, Arizona, January 7-12, 1980
TELESCOPES FOR THE 1990's

January 22, 1980

University of California

LOS ALAMOS SCIENTIFIC LABORATORY
Post Office Box 1663 Los Alamos, New Mexico 87545
An Affirmative Action/Equal Opportunity Employer

Form No. 836 R4
St. No. 2629
12/78

UNITED STATES
DEPARTMENT OF ENERGY
CONTRACT W-7405-ENG-36

GROUND BASED AUTOMATED TELESCOPE

Stirling A. Colgate
University of California
Los Alamos Scientific Laboratory
Los Alamos, NM 87545
and
New Mexico Institute of Mining and Technology
Socorro, NM 87801

and

William Thompson
New Mexico Institute of Mining and Technology
Socorro, NM 87801

January 22, 1980

ABSTRACT

We recommend that a ground-based automated telescope of the 2-meter class be built for remote multiuser use as a natural facility. Experience dictates that a primary consideration is a time shared multitasking operating system with virtual memory overlayed with a real time priority interrupt. The primary user facility is a remote terminal networked to the single computer. Many users must have simultaneous time shared access to the computer for program development. The telescope should be rapid slewing, and hence a light weight construction. Automation allows for the closed loop pointing error correction independent of extreme accuracy of the mount.

Before the advent of time shared computer operating systems accessed through CRT terminals, we were all used to submitting jobs as a card deck in the "batch" mode. Before that, only the engineer or operator had "hands on" control of the computer. It would seem that optical astronomy is just emerging from hands-on to batch mode, yet we will shortly operate an optical telescope in space in the remote time shared mode without yet having one ground based optical telescope operating in the same mode.

The criteria for a ground based automated telescope operating in the time shared mode are the following:

1. The telescope must be a National Facility with a major fraction of its time allocated by the standard procedure of porposal and review.
2. The optical data must be available to any observer transmitted by a national digital telecommunications network, e.g., Telenet.
3. The basic functions of the telescope must be under the observing program control through the primary computer.

The implementation of such a system is similar to our objective of a program for an automated search for supernovae. In the course of this work we have arrived at several convictions that may be useful to others.

First and foremost is the recognition that the software effort will be many times greater ($\sim \times 10$) than the hardware effort. (For example, the Space Shuttle is recognized as a hardware intensive project, yet the original contracts for its development included $\times 10$ the cost in software as in hardware.) A digital telescope should be no exception and indeed we have found that software is overwhelmingly the major effort. As a consequence, the operating system, computer, interface and telescope hardware should all be designed with this perception in mind.

If you have one task that is ten times larger than all others, it stands to reason that it is wise to have more than one person work on the problem simultaneously. This is why the operating system must be a multiuser virtual memory time shared system. Multiuser includes not only multiprogram access, but also multitasking where a separate task (like a user) is assigned to each of several telescope functions; e.g., telescope slewing, the digital cameras CCD or SIT star field pattern recognition, spectra, guiding, frame "taking", reduction and calibration, catalogue access and maintenance of stars, galaxies, quasars, and nebulae . A multitasking operating system allows all of these functions to be taking place concurrently provided a real time interrupt of the operating system can take place. Most multiuser operating systems assign a sequential time slice to each user. A telescope must be able to command attention with a priority above other tasks.

In addition there must be:

1. User observation program test, development and scheduling.
2. User data format and presentation.
3. User intervention.
4. Interlocks, telescope safety, and fault diagnosis.

A multiuser environment allows any task to talk to another task by passing parameters much like the function "mail" where any user passes messages to other users.

Many may ask why not use a separate microcomputer for each task? One then has the problem of networking all these computers - which can be done but it is presently much more difficult - again because the software effort is much greater than the hardware effort. It is much more efficient to time share one CPU that "knows" where everything (everyone) is located and where the operating system is a development for thousands of installations.

We have been modifying a Prime 300 (a 16-bit minicomputer with 128K core and 70 Mbyte disc) multiuser time sharing system to handle priority interrupts so that we can perform multitasking in a real time environment.

Computer System for Automated Control

Although adequate control of an automated telescope is a demanding job for both hardware and software, creating the software has proven to be the most difficult task (in terms of time and effort) for the Digitized Astronomy telescope. The size of the current software effort (on the order to 50,000 lines of FORTRAN source code) and the complexity of the task dictated most of the decisions when a minicomputer was chosen to replace the IBM 360/44 which formerly controlled the telescope.

The software system on the 360/44 was constructed as a series of overlays which were called into memory from disk storage as needed, replacing the former phase which had been executing. Data was passed from one phase overlay to another through FORTRAN named common areas which remained in memory between phases. Some phases had grown in size to as much as 200K bytes although most were about 120K bytes long. Much effort had been expended in creating and maintaining these overlays due to their conceptual complexity and to the means of passing the data. When data is passed in common, many phases have access to the data and tracing an error back through multiple branching data paths (and through data with different aliases in different phases) becomes incredibly complex.

At first sight, transferring this system to a minicomputer looked very bleak since most minicomputers have a limited address space of only 64K bytes.

To fit Digitized Astronomy overlays into this would require not only halving the phases, but usually quartering the common areas that still had to be resident in all phases. Indeed, in one phase in which the picture of 16K bytes (1 pixel/byte) was subtrated from a stored picture (also 16K butes) to produce a differenced picture (1 pixel/integer = 32K bytes) the whole address space might be used by the data alone!

Research into the current state of minicomputers soon revealed that we were not alone in this problem and that some manufacturers have gone to great efforts to solve it.

Two methods are perhaps well exemplified by Prime Computer machines and Interdata machines. The Interdata line of minicomputers evolved from 16 bit-machines with a limited address space to 32-bit machines with the ability to address a megabyte (1 MB) of main memory. Thus, if you have bigger programs you can buy more memory to run them in. The prime machines on the other hand, use virtual (paged) memory. Since most programs exhibit a high degree of linearity and localization it is practical to map a virtual address space to disk storage and page in only those sections of the program which are needed to execute the program at the current time. The ever increasing number of machines which use this technique testify to its utility and efficiency.

Neither of these machines (in their minimal configurations) were capable of supporting the Digitized Astronomy software without breaking it into overlays. We were also looking for time sharing capabilities, since the productivity of programmers, each with their own terminal is greatly increased over the complicated keypunching and updating of files to which we were accustomed. This concept of time sharing, or multitasking provided the clue for the next transformatting of the Digitized Astronomy system. The phase overlays which were so difficult to maintain were, in fact, seperate tasks (since this was

generally the most obvious way of breaking long programs into smaller ones.) As separate tasks, perhaps they could be run in a multitasking environment. Indeed, as this concept has been further expanded upon it has become apparent that much code was originally written to handle the synchronization between phases (tasks) that could be thrown away if this overhead is instead left to a multitasking operating system. There are many ways of handling multitasking. Consider, for example, four phases each 128K bytes long which have been converted to tasks (with appropriate intertask communications) running in an Interdata 7/32 with 256K bytes memory and in a Prime 300 with 128K bytes memory (actual machines proposed by vendors.) The multitasking on the Interdata will roll-in/roll-out tasks as space is needed to execute the current task, so if the tasks are executed sequentially after the second task has received it time slice every other time slice will require rolling out a task in memory and rolling in a new one from disk--256K bytes transfer per time slice! In a paged memory, however, the working set, i.e., necessary pages, for each task will reside in memory simultaneously (in practice, the working set for most programs on the Prime 300 seems to be about 10K-16K bytes). Then if a reference is made to a page not in memory, it will replace one in memory - a transfer of 2K bytes. Note this does not mean a page is swapped in and out for every instruction and in fact only a few or no swaps will be expected in any given time slice. A program which caused a swap with every instruction would surely be beyond comprehension of mere mortals.

Once we had decided that a virtual memory, multitasking system was by far the most appropriate for Digitized Astronomy some other interesting problems arose. The telescope is primarily a real-time device. Tasks on the other, operate in their own time, they exist only as long as they are being executed by the CPU. If we were to use programmed I/O as expected for the many 16-bit

data transfers, a task might query the telescope for some information but not be executing when the information returned! The solution, is to use Direct Memory Access along with drivers embedded in the operating system. The data buffers remain in the user address space and are locked into memory while the data transfer is taking place The CPU microcode is always ready to pause, even in the middle of an instruction to process data arriving via Direct Memory Access, thus ensuring that the data are never lost. The task requesting input from the telescope is put to "sleep," that is, it receives no further time slices until the data transfer has been completed.

Although it may appear inefficient at first, the overhead involved in using DMA instead of programmed I/O for small transfers is not too great and for large transfers is much more efficient. In addition, using the same method of transfer for all data made the design of the interface much simpler and easier to understand.

Another aspect of multitasking which was considered from the very start was intertask communication. Data have to be passed from task to task and there needed to be some means of synchronizing tasks. A common method of synchronization is to use semaphores with send and wait operations. For example, the telescope control task might execute a "wait on Semaphore A" instruction when it has finished moving the telescope and then go to sleep. The data collection task might then execute a "send to semaphore A" instruction when it is finished with the telescope. Other problems might require actual movement of data between tasks. For these a message transfer facility furnished by the operating system would be usually used. We have decided to modify the the Primus operating system to implement the message classes proposed by Greg Andrews (1976, in which both standard messages are used and zero-length messages serve as semaphores for synchronization. We are

thus still in the processes of establishing Digitized Astronomy on the Prime 300 computer and Primos III operating system, but we believe the major problems have been solved.

Some additional hardware considerations

The telescope should be relatively fast slewing so that the usual setting time (say for multiobject surveys) is small compared to the picture integration time. Since 10% photometry can be obtained of a 19th m object in $\lesssim$ 10 s with a 2 meter telescope, slewing should require no more than several seconds. This requires a light weight mirror, rigid mount, and very fast motor drive. Our own experience and calculations (Colgate, Moore, and Carlson 1975) say that one can design a central support, very light weight mirror, based on the homology principal of uniform stress (Van Hoerner 1967) that will maintain the necessary figure accuracy, but that pointing accuracy will be modestly compromised. One can either correct for pointing predictively by a complex distortion program, or "close the loop" as we do by recognizing a star field (usually an 8th m, bright star) and resetting the coordinates. This procedure of slew - take a picture, process it for several bright stars, recognize bright star, calculate pointing error, and finally correct coordinates requires several seconds and only needs to be done after telescope position changes of $\gtrsim$ 10 degrees. During supernova search operation, our past experience showed that each nearby field was adequately accurate to a few seconds of arc for a 1 degree slew so that only a periodic update of the coordinates was necessary. We digitally controlled stepping motors to slew the telescope through a gear drive. The criterion for speed is the number of steps per second since the minimum step size should be small ($\cong$ ½ arc/s for guiding and the maximum slew rate, 1 radian in 10 s, requires 4×10^4 steps/s. Each digital step of a stepping motor (8 per cycle) can be further divided

into (8 to 32) substeps by a digital to analog converter (Colgate et al 1970, Dittmar 1979) Our minimum step size is 1 arc/s so that our full step rate of 2 $\times 10^4$ steps/s is inherently capable of a stepping rat of $\sim 6 \times 10^5$ s^{-1}. We do not use the capability because our optics do not require it, i.e., a slit width of several arc seconds and a pixel size of 1 arc s. In such a system where an up down counter follows the stepping motor, one does not need accurate digital angular encoders. We use a simple micro switch (a one bit encoder) on each axis to initialize the up down counters and then further update the position on a bright star during initial setup. A relatively modest twelve bit encoder on each axis would be an advantage to recover from a "lost" condition without having to slew to the "microswitch" position.

Two values of F-number are useful but not necessary. For supernova searches, where nearby galaxies should be searched, the change in F number is required in view of the limited number of pixels of the digital TV detector. Similar to slewing, the F number change must be automated for change in several seconds.

We can focus the telescope remotely, but have not automated this function under computer control because of lack of need. We used a thermal compensating element in the mount that works better than expected and so refocus normally only several times a year. It would have been less hardware and more software to have automated the focus function.

Acknowledgements

We are indebted to many, among whom are Bryan Edwards and Richard Carlson.

This work was partially supported by the NSF Astronomy Section and by the Department of Energy.

References

Andrews, Gregory, 1976, Message Classes: An Approach to Process Synchronization," a Technical Report No. 76-275 from the Dept. of Computer Science, Cornell University, Ithaca, NY.

Colgate, S. A. Moore, E. P., and Carlson, R., 1975, "A Fully Automated Digitally Controlled 30-Inch Telescope," Pub. of ASP 77, 565.

Ditmar, D. N., McDonald Observatory, University of Texas, private communication, 1980.

Van Hoerner, 1967, "Design of Large Steerable Antennas," Astronomical J., 72, 35.

QUESTIONS:

Name: Jerry Nelson
Question: Why do you desire the ability to change f ratios very rapidly?
Answer: Replied in text.

Name: Labeyrie
Question: Have you considered long distance links, like from here to Hawaii or Chili?
Answer: Telenet costs about 5¢ per minute to support a terminal anywhere within the US. It should be 2 to 3 times this around the world.

Name: G. Gabor
Question: What do you mean by a time shared telescope? What would be the turnaround time for different observers?
Answer: Replied in text.

Name: Dave Cudaback
Question: What communication band width is required between telescopes and remote users?
Answer: It requires roughly 1200 band to support a terminal - 5¢ per minute Telenet.